Supplementary workflow

Greg Gloor

17 January, 2018

Contents

About this document

This document is an .Rmd document and can be found at:

github.com/ggloor/templates

The document is a template for two column R markdown. It requires an installation of LATEXto work properly. This document can contain interspersed markdown and R code that may be compiled into a pdf document and supports the figures and assertions in the main article. R code is not exposed in the pdf document but is referred to by `R code block` in the text so that the interested reader can work through the example code themselves.

Reproducing the analysis

From an R command prompt you can compile this document into PDF if you have LATEXand pandoc installed:

`rmarkdown::render('two_column.Rmd')` or you can open the file in RStudio and compile in that environment.

R packages required

We will need the following R packages and add-ons (`R_block_1`).

1. knitr (CRAN)

Common data transforms in high throughput sequencing

Fundamentally, the goal of any experiment is to determine something about the environment that was sampled. After all, we are attempting to use HTS to determine something of interest about the underlying environment. Thus, we need to have some equivalence between the samples before sequencing and the samples after sequencing. The simplest case would be that there would be a linear relationship between the data that we could obtain from the environment, and the data that was actually collected by HTS.

We can think about the underlying data on a univariate basis; do the features across all samples follow a Gaussian distribution? or do they follow some unknown distribution? If so, can we transform the data to approximate a Gaussian distribution? This mode of thinking leads to the use of square-root, arcsine or Hellinger transformations since they appear to transform the data into a distribution that can be interpreted. However, as we shall see below, none of these univariate transformations is suitable.

It is more desirable to think about HTS data in a multivariate way as a 'composition' because the total count of molecules in the underlying sample (the environment) is always a confounding variable (Lovén et al., 2012). This way of thinking led to multivariate data normalizations.

Notation

We use the following notation throughout. Column vectors contain samples $\vec{\mathbf{s}}$ and row vectors contain features $\vec{\mathbf{f}}$. There are D features and n samples, thus the data are contained in matrix $M = D \times n$. The j^{th} sample is denoted as s_j, the i^{th} feature of all samples is denoted as s_{i-}, and the value for the i^{th} feature of the j^{th} sample is referred to as s_{ij}.

Simple proportional type transformations

The simplest normalization is to determine the relative abundance (rAB), or proportion, of the i^{th} feature in a sample as in Eq. 1.

$$rAB_i = \frac{s_i}{\sum \vec{\mathbf{s}}} \tag{1}$$

This normalization is also referred to as the total sum scaling (TSS) normalization. The rAB measure requires only the read count observed for a the feature s_i and the total read count of the sample $\sum \vec{\mathbf{s}}$. Since this measure is generally skewed, it is often log-transformed prior to analysis.

A further normalization was proposed early in the RNA-seq field where the reads per kilobase per million mapped (RPKM)(Mortazavi et al., 2008) method was used initially to place the read counts for each feature within and between samples on a common scale.

For this we also needed to know a scaling factor K, and the length of the feature L_i; from this, the RPKM value for the i^{th} feature for each sample was calculated as in Eq. 2.

$$RPKM_i = \frac{(K \cdot C_i)}{\sum C \cdot L_i} \tag{2}$$

When the equation is placed in this form it is obvious that RPKM is simply a scaled rAB where each rAB value is divided by its length multiplied by a constant. In compositional terms, RPKM is an unclosed perturbation of the original data.

Further research suggested that RPKM was not appropriate for comparison of features between samples. The goal of RPKM was to 'count' reads per feature per cell. In the original paper the authors supplied an equivalence and an RPKM value of 1 RPKM equalled one transcript in each cell in the C2C12 cell line, but in liver cells, a value of 3 RPKM equalled one transcript per cell. Thus, from the start, this normalization was unable to normalize between-condition read counts.

The transcripts per million (TPM) normalization was advocated next (Li et al., 2010). Patcher (Pachter, 2011) showed the equivalence between RPKM and TPM, and in compositional terms TPM is simply a compositionally closed form of RPKM multiple by a constant as in Eq. 3.

$$TPM_i = \frac{RPKM_i}{\sum RPKM} \cdot K \tag{3}$$

The rAB, RPKM and TPM normalizations are thus all very similar, differing only in the scaling of individual features, and do not allow normalization between conditions unless the conditions contain the same input number of RNA molecules. In a very real sense, these normalizations deliver proportional data, scaled or perturbed to make the data appear as if they are numerical, and not proportional.

A related transformation is 'rarefaction' or subsampling without replacement to a defined per-sample read count. This transformation was widely used in the 16S rRNA gene sequencing field. Rarefaction to a common read count gives a composition, that is scaled such that low count features often are replaced by 0 values (McMurdie and Holmes, 2014). For this reason, rarefaction has now been largely replaced with the median of ratios method described below.

The median of ratios count normalization

Further work found that none of these methods were appropriate, since the read count per sample continued to confound the analyses (Lovén et al., 2012). Thus, the scaling normalization methods were proposed (Robinson and Oshlack, 2010). There are two main scaling normalizations, but both operate on the common assumption that by normalizing all counts in a sample to a midpoint of each sample that the normalization can impute the *number* of each feature in the environment. The approaches differ largely in how the midpoint is determined. The median of ratios method (MR) is instantiated in DESeq2 (and others), and the trimmed mean of M values (TMM) method is used by edgeR (and others). The DM method will be demonstrated and used, but the TMM gives substantially similar results, and uses the same basic logic since values are scaled by a feature-wise midpoint.

The DM method calculates the ratio of the features to the geometric mean, G_i, of each feature across all samples, and then takes as the normalization factor the median ratio per sample as the scaling factor. Each feature is then divided by the scaling factor to place each sample on an equivalent count scale. The idea is that the DM normalization 'opens' the data from being compositional to being scaled counts. As we shall see, it is impossible to open the data, and while the scaled counts have some useful properties, removing compositional constraints are not among them.

The multi-step normalization MR normalization attempts to normalize for sequencing depth thus 'opening' the data, and proceeeds as in the multistep Eq. 4. Here we start with two sample vectors $\vec{\mathbf{s}}_1$ and $\vec{\mathbf{s}}_2$, and calculate a vector of geometric means of the features $\vec{\mathbf{g}}$. Ratio vectors, $\vec{\mathbf{r}}_j$ are calculated by dividing the sample vectors by the geometric mean vector, and the median of the ratio vectors is determined. Finally, the sample vectors are divided by the median of the ratio vector for each sample.

$$\begin{aligned}\vec{\mathbf{g}} &= \mathrm{G}_{i-} \\ \vec{\mathbf{r}}_j &= \vec{\mathbf{s}}_j / \vec{\mathbf{g}} \\ \vec{\mathbf{d}}_j &= \vec{\mathbf{s}}_j / Md(\vec{\mathbf{r}}_j)\end{aligned} \tag{4}$$

In Table 1 we can see that the median ratio for each sample $\vec{\mathbf{r}}_j$ samples may be different in each sample, and that the particular feature that is the median may itself be different, the median feature is in boldface in the table. Thus, by construction the feature values in each sample can be scaled by different amounts in each sample.

Table 1: Example calculation of DM normalization

Part	$\vec{\mathbf{s}}_1$	$\vec{\mathbf{s}}_2$	$\vec{\mathbf{g}}$	$\vec{\mathbf{r}}_1$	$\vec{\mathbf{r}}_2$	$\vec{\mathbf{d}}_1$	$\vec{\mathbf{d}}_2$
F1	1500	1000	1224.7	1.22	**0.81**	1219.5	1234.6
F2	25	15	19.4	1.29	0.77	20.3	18.5
F3	1000	500	707.1	1.41	0.71	813.0	617.3
F4	75	50	61.2	**1.23**	0.82	61.0	61.7
F5	500	1500	866.0	0.58	1.73	406.5	1851.9

Log-ratio transformations

There are three main log-ratio transformations; the additive log-ratio (alr), centred log-ratio (clr) and the isometric log-ratio (ilr) [Pawlowsky-Glahn et al. (2015)}.

Using the same notation as above for a sample vector $\vec{\mathbf{s}}$ of D 'counted' features (taxa, operational taxonomic units or OTUs, genes, etc.) $\vec{\mathbf{s}} = [s_1, s_2, ...s_D]$:

The alr is the simply the elements of the sample vector divided by a presumed invariant feature, which by convention here is the last one:

$$\begin{aligned}\vec{\mathbf{x}}_{alr} = [&log(x_1/x_D), log(x_2/x_D), \\ &\ldots log(x_D - 1/x_D]\end{aligned} \tag{5}$$

This is similar to the concept used in quantitative PCR, where the relative abundance of the feature of interest is divided by the relative abundance of a (presumed) constant 'housekeeping' feature. Of course there are two major drawbacks. First, that the experimentalist's knowledge of which, if any, features are invariant is necessarily incomplete. Second, is that the choice of the (presumed) invariant feature has a large effect on the result if the presumed invariant feature is not invariant, or if it is correlated with any other features in the dataset. Interestingly, an early proposal was to use the geometric mean of a number of internal controls (Vandesompele et al., 2002), leading to the next transformation.

The centered log-ratio (clr) transformation introduced by [Aitchison (1983)},[Aitchison (1986)} uses the geometric mean of all features as the denominator:

$$\begin{aligned}\vec{\mathbf{x}}_{clr} =&[log(x_1/\mathrm{G}(\vec{\mathbf{x}})), \\ &log(x_2/\mathrm{G}(\vec{\mathbf{x}})), \\ &\ldots log(x_D/\mathrm{G}(\vec{\mathbf{x}}))]\end{aligned} \tag{6}$$

where $\mathrm{G}(\vec{\mathbf{x}}) = \sqrt[D]{x_1 \cdot x_2 \cdot ... \cdot x_D}$, the geometric mean of $\vec{\mathbf{x}}$.

The clr is often criticized since it has the property that the sum of the clr vector must equal 0. This constraint causes a singular covariance matrix; i.e., the sum of the covariance matrix is always a constant (Pawlowsky-Glahn et al., 2015). However the clr has the advantage of being readily interpretable, a value in the vector is its abundance *relative* to a mean value.

The ilr is the final transformation, and is a series of sequential log-ratios between two groups of features. For example, the philr transformation is the series of ratios between OTUs partitioned along the phylogenetic tree (Silverman et al., 2017), although any other sequential binary partitioning scheme is also possible (Pawlowsky-Glahn et al., 2015). The ilr transformation does not suffer the drawbacks of either the alr or clr, but does not allow for insights into relationships between single features in the dataset. Nevertheless, ilr transformations permit the full-range of multivariate tools to be used, and are recommended whenever possible.

The ilr and clr are directly comparable in a two important ways: First, the distances between samples computed using an ilr and clr transformation are equivalent. Second, the clr approaches the ilr in other respects as the number of features becomes large. In this respect, the large number of features—hundreds in the case of OTUs, thousands in the case of genes—in a typical experiment works in our favour. Thus, while not perfect, the clr is the most widely used transformation. However, care must be taken when interpreting its outputs since single features must always be interpreted as a ratio between the feature and the denominator used for the clr transformation. The problems of using clr are apparent when some subcomposition or group of taxa is analysed for further insight since the geometric mean of the subcomposition is not necessarily equal to that of the original composition, leading to potential inconsistencies.

Log-ratio values of any type do not need to be normalized since the total sum is a term in both the numerator and the denominator. Thus, the same log-ratio value will be obtained for the vector of raw read counts, or the vector of normalized read counts, or the vector of proportions calculated from the counts. Thus, log-ratios are said to be equivalence classes such that there is no information in the total count (aside from precision) (Barceló-Vidal et al., 2001).

Attempts to 'open' the data are doomed to failure because the data cannot be moved from the simplex to Euclidian space. The total count delivered by the sequencing instrument is a function of the instrument and not the number of molecules sampled from the environment, thus the total count has no geometric meaning. If the data are collected in such a way that the total count represents the actual count in the environment, then the data are not compositional and issues regarding compositional data disappear. However, at present all sequencing platforms deliver a fixed-sum, random sample of the proportion of molecules in the environment.

Note that this does not mean that the read depth is irrelevant since more reads for a sample translate into greater precision when estimating the proportions (Fernandes et al., 2013).

Distance metrics

The microbiome and transcriptome literature are replete with distance metrics, and it is common to find that a single study will use several distance metrics to report their findings. This is a problem since it shows that practitioners are unsure of the reason to use a metric, and the use of more than one metric leads to data dredging and research degrees of freedom—both of which increase the chances of finding false positives in the data to a surety.

Distance metrics can be broadly divided into those that require partitioning and those that do not. The UniFrac (Lozupone and Knight, 2005; Lozupone et al., 2011) and philr (Silverman et al., 2017) both require a phylogenetic tree, making these metrics applicable only to situations where the features can be so partitioned. For example, these distances are useful when examining 16S rRNA gene sequencing experiments. We have found that the unweighted UniFrac method is unreliable, and should be used with caution [Wong et al. (2016)}, a point that was made in the original UniFrac paper and subsequently forgotten. The philr metric is a drop-in replacement for the weighted UniFrac distance metric and should be used whenever possible, since `philr` is an ilr transformation of the data where the sequential binary partitions are made along the phylogenetic tree. The `philr` transformation is thus compositionally appropriate. In practice, the weighted UniFrac distance metric provides similar results to the Aitchison distance, described below, and the ilr distance calculated using the philr transform approaches the Aitchison distance when the number of features is large.

Several non-phylogenetic distances are in widespread use in the literature. These will be discussed in turn below, and their effects on distances between a random samples illustrated.

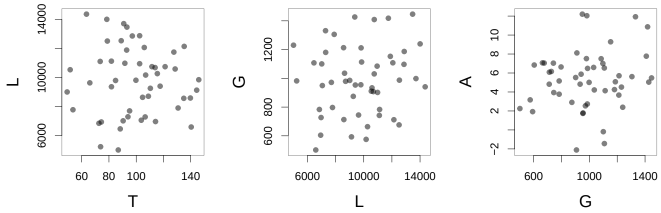
Distances in counts

Ideally, we use distance metrics to inform us as to something of relevance in the actual sample. That is, if we collect our data on the numbers of tigers, ladybugs, gnus and space aliens, what can we infer about the actual data *after the transformation*? That is, what is the correspondence between distances in the underlying data and in the data after transformation?

At this point we will set up a random dataset, composed of four features (T, L, G, A) and 50 random samples with mean values of 100 tigers, 10000 ladybugs, 1000 gnus and 5 space aliens. The features will be drawn from a Normal distibution, although a random uniform distribution or any other distribution will give the same results. We are not, at this point, attempting to mimic a distribution found in a real dataset, but are instead showing the general properties of the distance metrics.

```
set.seed(13)
T <- rnorm(50, mean=100, sd=25)
L <- rnorm(50, mean=10000, sd=2500)
G <- rnorm(50, mean=1000, sd=250)
A <- rnorm(50, mean=5, sd=2.5)
rand.data <- cbind(T,L,G,A)
par(mfrow=c(1,3), pch=19, col=rgb(0,0,0,0.5),
    cex=1.5, cex.lab=1.5)

plot(T,L)
plot(L,G)
plot(G,A)
```



Plotting three of the possible combinations, we can see that the features are essentially uncorrelated with each other and each sample is a random distances from any other. Any inference we make from transformations of this data must be relatable to this ‘ground truth’. I now run through each of the transformations in turn, and illustrate the difference between the actual data, and the transformed data.

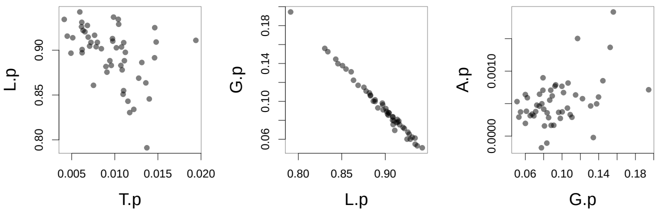
Euclidian Distance of TSS scaling transformation

The TSS scaling transformation is simply a conversion of each sample from a count to a proportion.

```
rand.data.prop <- t(apply(rand.data, 1,
    function(x) x/sum(x)))

par(mfrow=c(1,3), pch=19, col=rgb(0,0,0,0.5),
    cex=1.5, cex.lab=1.5)

plot(rand.data.prop[,"T"],rand.data.prop[,"L"],
    xlab="T.p", ylab="L.p")
plot(rand.data.prop[,"L"],rand.data.prop[,"G"],
    xlab="L.p", ylab="G.p")
plot(rand.data.prop[,"G"],rand.data.prop[,"A"],
    xlab="G.p", ylab="A.p")
```



By comparing the TSS transformed data to the non-transformed data, we can see that the structure of the data itself has changed dramatically. The two most abundant features, G and L, which are uncorrelated in the actual data are now almost perfectly negatively correlated when the same data are converted to proportions. This is because the data are now not real numbers, but are instead proportions and are constrained by the arbitrary sum of 1: *the data are now compositional data.*

Euclidian Distance of DM transformation

Bray-Curtis Dissimilarity

Jensen-Shannon Divergence

Aitchison Distance

Transformations and distance

References

Aitchison, J. (1983). Principal component analysis of compositional data. *Biometrika* 70, 57–65.

Aitchison, J. (1986). *The statistical analysis of compositional data.* London, England: Chapman & Hall.

Barceló-Vidal, C., Martín-Fernández, J. A., and Pawlowsky-Glahn, V. (2001). “Mathematical foundations of compositional data analysis,” in *Proceedings of IAMG*, 1–20.

Fernandes, A. D., Macklaim, J. M., Linn, T. G., Reid, G., and Gloor, G. B. (2013). ANOVA-like differential expression (aLDEx) analysis for mixed population rNA-seq. *PLoS One* 8, e67019. doi:10.1371/journal.pone.0067019.

Li, B., Ruotti, V., Stewart, R. M., Thomson, J. A., and Dewey, C. N. (2010). RNA-seq gene expression estimation with read mapping uncertainty. *Bioinformatics* 26, 493–500. doi:10.1093/bioinformatics/btp692.

Lovén, J., Orlando, D. A., Sigova, A. A., Lin, C. Y., Rahl, P. B., Burge, C. B., et al. (2012). Revisiting global gene expression analysis. *Cell* 151, 476–82. doi:10.1016/j.cell.2012.10.012.

Lozupone, C., and Knight, R. (2005). UniFrac: A new phylogenetic method for comparing microbial communities. *Applied and environmental microbiology* 71, 8228–8235.

Lozupone, C., Lladser, M. E., Knights, D., Stombaugh, J., and Knight, R. (2011). UniFrac: An effective distance metric for microbial community comparison. *ISME J* 5, 169–72. doi:10.1038/ismej.2010.133.

McMurdie, P. J., and Holmes, S. (2014). Waste not, want not: Why rarefying microbiome data is inadmissible. *PLoS Comput Biol* 10, e1003531. doi:10.1371/journal.pcbi.1003531.

Mortazavi, A., Williams, B. A., McCue, K., Schaeffer, L., and Wold, B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nat Methods* 5, 621–8. doi:10.1038/nmeth.1226.

Pachter, L. (2011). Models for transcript quantiffication from RNA-seq. *ArXiv* 1104.3889.

Pawlowsky-Glahn, V., Egozcue, J. J., and Tolosana-Delgado, R. (2015). *Modeling and analysis of compositional data*. John Wiley & Sons.

Robinson, M. D., and Oshlack, A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol* 11, R25.1–R25.9. doi:10.1186/gb-2010-11-3-r25.

Silverman, J. D., Washburne, A. D., Mukherjee, S., and David, L. A. (2017). A phylogenetic transform enhances analysis of compositional microbiota data. *Elife* 6, 21887. doi:10.7554/eLife.21887.

Vandesompele, J., De Preter, K., Pattyn, F., Poppe, B., Van Roy, N., De Paepe, A., et al. (2002). Accurate normalization of real-time quantitative rT-pCR data by geometric averaging of multiple internal control genes. *Genome Biol* 3, RESEARCH0034.

Wong, R. G., Wu, J. R., and Gloor, G. B. (2016). Expanding the UniFrac toolbox. *PLoS One* 11, e0161196. doi:10.1371/journal.pone.0161196.